Oracle Financial Services Compliance Studio

Architecture Guide

Release 8.1.2.5.0

May 2024

F48788-01

ORACLE

Financial Services

OFS Compliance Studio Architecture Guide

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Document Control

Table 1 lists the document control of this guide:

Table 1: Document Control

Revision Number	Revision Date	Change Log
8.1.2.5.8	May 2024	Added the following sections: - High Availability Configuration - Compliance Studio with Multiple PGX Servers (Using Load Balancer) - Setup Disaster Recovery (DR) in Compliance Studio - Frequently Asked Questions (FAQs)
8.1.2.5.0	July 2023	Added a new note for Python Interpreter in the Component Details section. The support for Legacy ETL is deprecated in the current release, and the related note is added in the required sections.
8.1.2.4.0	March 2023	Updated document to reflect that OpenSearch has replaced Elastic Search in 8.1.2.4.0. Removed ORE Interpreter in the Component Details section. Removed Cloud Architecture diagram in the Simplify Architecture section. Added graph service, pipeline service, and Scenario Conversion Utility in the Component Details section. Updated architecture diagram in the following sections: - Architecture Overview - Use Cases
8.1.2.3.0	January 2023	Updated for the version v8.1.2.3.0 release.
8.1.2.1.0	January 2023	Updated Architecture diagram in the following sections: - Native Architecture - Scenario Authoring - Investigation Hub Updated Notebook Server UI component in the Component Details section.
8.1.2.1.0	September 2022	Updated for the version v8.1.2.1.0 release.
8.1.2.0.0	March 2022	Updated Architecture Overview and Component Details the version v8.1.2.0.0 release.
8.1.1.1.0	November 2021	Updated for the version v8.1.1.1.0 release.
8.1.1.0.0	October 2021	This is created for the version v8.1.1.0.0 release.

Table of Contents

1 Preface

This preface provides information for the Oracle Financial Services Compliance Studio (OFS Compliance Studio) Architecture Guide.

Topics:

- About this Guide
- Audience
- Related Documents
- Conventions
- Abbreviations

1.1 About this Guide

This document provides the architecture details and the key components of OFS Compliance Studio. In addition, it also describes the application authentication process and use cases.

1.2 Audience

Oracle Financial Services Compliance Studio Architecture Guide is intended for implementation consultants and administrators who can view the high-level architecture of the Compliance Studio solution.

1.3 Related Documents

You can access the following additional documents related to the OFS Compliance Studio application from the Oracle Help Center (OHC) Documentation Library.

- Oracle Financial Services Compliance Studio Installation Guide
- Oracle Financial Services Compliance Studio Administration and Configuration Guide
- Oracle Financial Services Compliance Studio User Guide
- Oracle Financial Services Compliance Studio Matching Guide
- Oracle Financial Services Compliance Studio Data Model Guide

To find additional information about how Oracle Financial Services solves real business problems, see our website.

1.4 Conventions

Table 2 explains the text conventions used in this guide.

Table 2: Document Conventions

Convention	Description
Italics	Names of books, chapters, and sections as references
Bold	Emphasis and need for attention
Hyperlink	Hyperlink type indicates the links to external websites, and internal document links to sections.

1.5 Abbreviations

Table 3 lists the abbreviations used in this document.

Table 3: Abbreviations

Abbreviation	Meaning
OFS	Oracle Financial Services
OFSAAI	Oracle Financial Services Analytical Applications Infrastructure
OHC	Oracle Help Center
MOS	My Oracle Support
OFSAA	Oracle Financial Services Analytical Application
BD	Behavior Detection
FCDM	Financial Crime Data Model
MMG	Model Management and Governance
SSO	Single Sign-On
SSH	Secure Shell
OOB	Out of the Box
PGX	Parallel Graph Analytics
AML	Anti-Money Laundering
ML	Machine Learning
ML4AML	Machine Learning for AML
ORE	Oracle R Enterprise
SAML	Security Assertion Markup Language
AAI	Advanced Analytics Infrastructure
HTTP	Hypertext Transfer Protocol
HTTPS	HTTP over SSL or HTTP Secure
SSL	Secure Socket Layer
TLS	Transport Layer Security
ETL	Extract, Transform and Load
SSH	Secure Shell Protocol
UI	User Interface
IDP	Identity Provider
REST	Representational State Transfer
GER	Global Entity Resolution
LDAP	Lightweight Directory Access Protocol

Table 3: Abbreviations

Abbreviation	Meaning
SID	System Identifier
REPL	Read Eval Print Loop
HA	High Availability
LB	Load Balancer

2 OFS Compliance Studio Architecture

This chapter focuses on the following architecture, components, and use cases.

Topics:

- Introduction
- Architecture Overview
- Components
- Component Details
- Communication Details
- Application Deployment
- Application Authentication
- Use Cases

2.1 Introduction

OFS Compliance Studio is an advanced analytics application that supercharges anti-financial crime programs for better customer due diligence, transaction monitoring, and investigations by leveraging the latest innovations in artificial intelligence, open-source technologies, and data management.

It combines Oracle's Parallel Graph Analytics (PGX), Machine Learning for AML, Entity Resolution, notebook-based code development, and enabling Contextual Investigations in one platform with complete and robust model management and governance functionality.

2.2 Architecture Overview

This topic provides the architecture details.

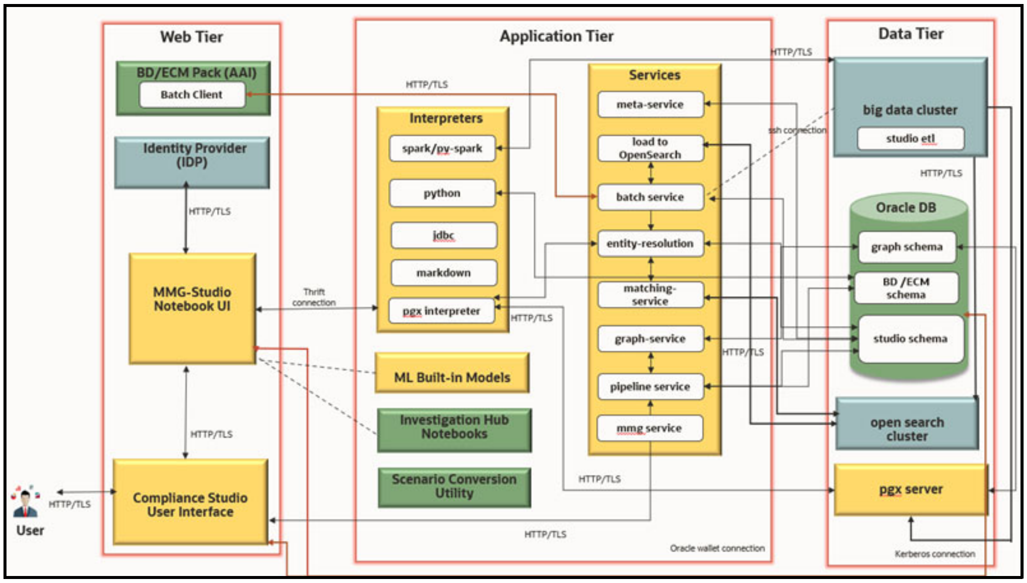
Topics:

- Native Architecture
- Simplify Architecture

2.2.1 Native Architecture

The following diagram exhibits the complete architecture of OFS Compliance Studio.

Figure 1: OFS Compliance Studio Complete Architecture



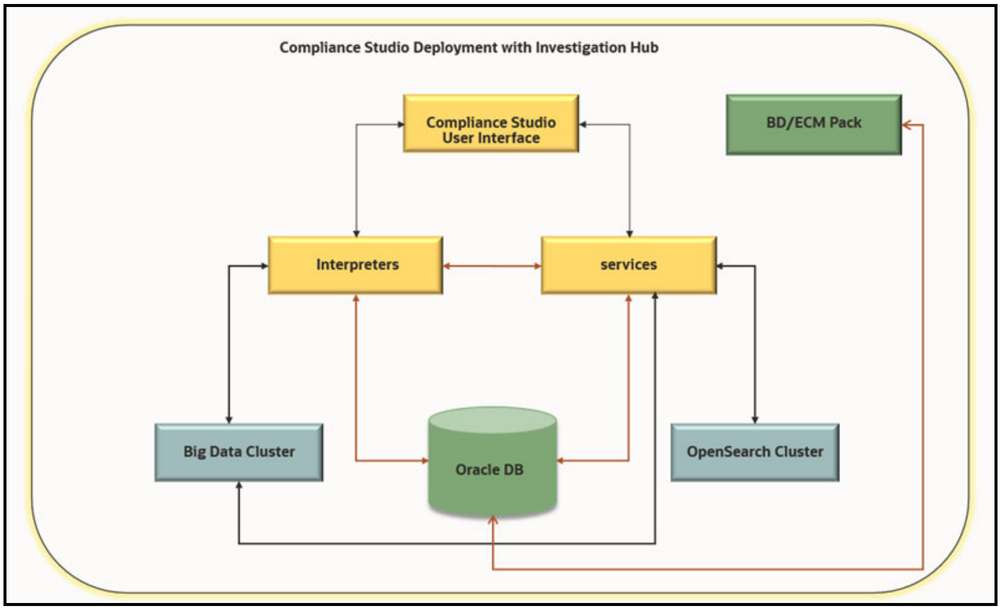
NOTE

- Compliance Studio components (indicated in the yellow color) are deployed on the same server.
- PGX Server can be deployed on the same server or other server based on Graph Sizing requirement.

2.2.2 Simplify Architecture

The following diagram exhibits the Simplified architecture of OFS Compliance Studio.

Figure 2: Compliance Studio Deployment with Investigation Hub


Compliance Studio Deployment with Investigation Hub
Compliance Studio User Interface
BD/ECM Pack
Interpreters
services
Big Data Cluster
Oracle DB
OpenSearch Cluster

2.3 Components

This topic provides a list of key components and third-party components.

Topics:

- Key Components
- Other Oracle Components
- Third-party Components

2.3.1 Key Components

The following components are bundled in the OFS Compliance Studio installer:

- OFS Compliance Studio Front End Service
  - Compliance Studio UI
  - Notebook Server UI
- OFS Compliance Studio Back End Service
  - Interpreters
  - Services
  - MMG Service
- ML4AML Models
- Python

- Parallel Graph Analytics Server

2.3.2 Other Oracle Components

- Behavior Detection (DB) Pack
- Enterprise Case Management (ECM) Pack
- Investigation Hub
- Scenario Conversion Utility
- Oracle DB

2.3.3 Third-party Components

- OpenSearch Cluster
- Big Data Cluster
- Identity Provider (IDP)

2.4 Component Details

Table 4 shows the component details.

Table 4: Component Details

Component/Service	Details
OFS Compliance Studio Front End Service	
Compliance Studio UI	You can access the Compliance Studio UI via browser and enter the login credentials along with the language. For valid login credentials, it navigates to the Workspace Summary page.
Notebook Server UI	You can access Notebook Server UI through the Compliance Studio UI.
OFS Compliance Studio Back End Service - Interpreters	
Spark Interpreter	You can connect to a big data cluster and create the models to perform analytics on data present in the Big data clusters.

Table 4: Component Details

Component/Service	Details
Python Interpreter	You can create/execute the Python models using this Interpreter. Analytics can be done with any python library. By default, python interpreters are configured with the following Virtual Environment: ● fcc-python ● fcc-python-ml4aml ● fcc-python-sane For more information, see the OFS Compliance Studio Installation Guide. **NOTE:** **From 8126, virtual environments will no longer be shipped with Compliance Studio and instead conda environment functionality will be used.** **These python interpreters (fcc-python, fcc-python-ml4aml, and fcc-python-sane) will be deprecated and removed in the future release but will remain in case of upgrades.**
JDBC Interpreter	You can create/execute the SQL models using this Interpreter. By default, this is connecting to Oracle Studio schema. You can connect to any schema by changing the interpreter configuration. For example, BD or ECM schemas
PGX Interpreter	● **pgx-java**: Java-based Interpreter, you can create/execute Java-based models and interact with the PGX server for graph analytics ● **pgql**: SQL is like an interpreter to query on the graph ● **pgx-python**: python based Interpreter with a PGX python client embedded in it to query on graph present in the PGX server. By default, this Interpreter points to ml4aml Python Virtual environment. ● **pgx-algorithm**: Graph toolkit that provides a graph query language and optimized analytics algorithms. For more information, see the website.
OFS Compliance Studio Back End Service	
Meta Service	This service is responsible for setting up metadata related to Compliance Studio in Studio schema.
Load to OpenSearch	This service manages OpenSearch indexes used to resolve the entity based on the matching rules.
Batch Service	This service is responsible for executing some of the batch jobs of Compliance Studio. For example, ETL for graph analytics or entity resolution

Table 4: Component Details

Component/Service	Details
Entity Resolution	It is responsible for resolving entities using matching and merging rules. ● **Graph ER**: It creates Similarity Edges between nodes by comparing the attributes of the nodes and identifying where the similarity is significant enough to create an edge so the nodes are linked with the graph model and can be analyzed as a single entity. ● **Global Party ER**; It creates a Global Party of similar entities by comparing multiple attributes of entities using matching and merging rules. For more information on merging and matching rules, see OFS Compliance Studio Matching Guide.
Matching Service	It is responsible for scoring in ER based on matching rules. It has the following scoring methods: ● Jaro-Winkler ● ML-Boosted Name ● ML-Boosted Address ● Levenshtein ● Individual ● Name ● Entity ● Name ● Default For more information on merging and matching rules, see the OFS Compliance Studio Matching Guide.
Graph Service	This service is used for managing graphs in Compliance Studio.
Pipeline Service	This service is used for extract transform and load data into target tables.
ML4AML Models	
ML Model Templates	The prepackaged Model templates allow you to perform the following: ● Model segmentation (model grouping) ● Load and Preview data ● User-defined transformation (deriving additional features) ● EDA ● Feature selections ● Model training and ● Evaluation ● Model Agnostics (Explainability) ● On-going validations
Python	Python contains all packages required to execute ML4AML models. For example, scikit-learn pandas
MMG Service	

Table 4: Component Details

Component/Service	Details
MMG Service	This service is used to manage the following functions: ● Workspaces and sandbox ● Data sources (external, local file, relational, and distributed) ● Model complete lifecycle, governance, and execution ● Batch and Scheduler services ● User roles and accesses ● User Provisioning and authentication
Other Oracle Components	
Parallel Graph Analytics Server	Graph analysis lets you reveal latent information that is not directly apparent from fields in your data but is encoded as direct and indirect relationships - metadata - between elements of your data. This connectivity-related information is not apparent to the naked eye but can have tremendous value when uncovered. PGX is a toolkit. For graph analysis, It supports both efficient graph algorithms and fast SQL-like graph pattern matching queries. FCGM is loaded into the PGX server for analytics.
BD PACK	In Compliance Studio, the graph model is based on the BD Pack's FCDM model and ML4AML using the same data model. For more information, see the Behavior Detection Application Pack.
ECM PACK	In Compliance Studio, the graph model is based on the ECM Pack's FCDM model. ECM is also used to correlate events generated via Compliance Studio and for case investigation. For more information, see the Enterprise Case Management Application Pack.
Oracle DB	Compliance Studio stores the metadata in the Oracle DB.
Investigation Hub	OFS Investigation Hub is an application built on Compliance Studio, allowing investigators to view the case and adhoc information within, then creates case narratives and insights, highlight risk factors and red flags meaningful to the investigation, and recommend actions based on graph scoring algorithms. For more information, see the Investigation Hub Application Pack.
Scenario Conversion Utility	This utility converts the Behavior Detection scenario into Compliance Studio scenario.
Third-party Components	
Identity Provider	Identity Provider (IdP or IDP) is required for SSO/SAML authentication.

Table 4: Component Details

Component/Service	Details
Big Data Cluster	Big Data Clusters allow you to deploy scalable clusters of SQL Server, Spark, and HDFS. These components run side-by-side to enable you to read, write, and process big data from Transact-SQL or Spark, allowing you to easily combine and analyze your high-value relational data with high-volume big data. It is used for ETL, a job that is converting our FCDM to FCGM. For more information on Technology Stack Matrices, see Oracle Financial Services Analytical Applications 8.1.2.0.0 Technology Matrix. **NOTE:** **This service is deprecated in the current release and will be removed in the future release.**
OpenSearch Cluster	An OpenSearch cluster is a group of nodes that have the same cluster name attribute. As nodes join or leave a cluster, they reorganize to evenly distribute the data across the available nodes. If you are running a single instance of OpenSearch, you have a cluster of one node. It is used for a matching service engine used for Entity Resolution and Similarity Edge for Graph Nodes.

2.5 Communication Details

Table 5 shows the communication details.

Table 5: Communication Details

Connection/Interface	Details
HTTP	Hypertext Transfer Protocol (HTTP) is a communication protocol in the application.
HTTPS	HTTPS (HTTP over SSL or HTTP Secure) uses a Secure Socket Layer (SSL), a secure protocol that works on top of HTTP to provide security. That means SSL encrypted data will be routed using protocols like HTTP for communication.
TLS	Transport Layer Security (TLS) encrypts data for private and sensitive information such as passwords, credit card numbers, and personal correspondence in the application.
ETL	Extract Transfer and Load (ETL) is the procedure of copying data from one or more sources into a destination system that represents the data differently from the source or in a different context. Data movement and graph loading is performed using ETL. **NOTE:** **This connection is deprecated in the current release and will be removed in the future release.**
Thrift connection	Thrift supports clean abstractions and implementations for data transport, data serialization, and application-level processing.

Table 5: Communication Details (Continued)

Oracle Wallet connection	Oracle Wallet is a file that stores database authentication and signing credentials. It allows users to securely access databases without providing credentials to third party software and quickly connects to Oracle products.
SSH connection	Secure Shell Protocol (SSH) hosts multiple channels simultaneously and transfers data in both directions.

2.6 Application Deployment

A separate installer is provided for the On-premise deployment.

For more installation information, you can see the respective OFS Compliance Studio Installation Guides.

2.7 Application Authentication

This topic provides the authentication details.

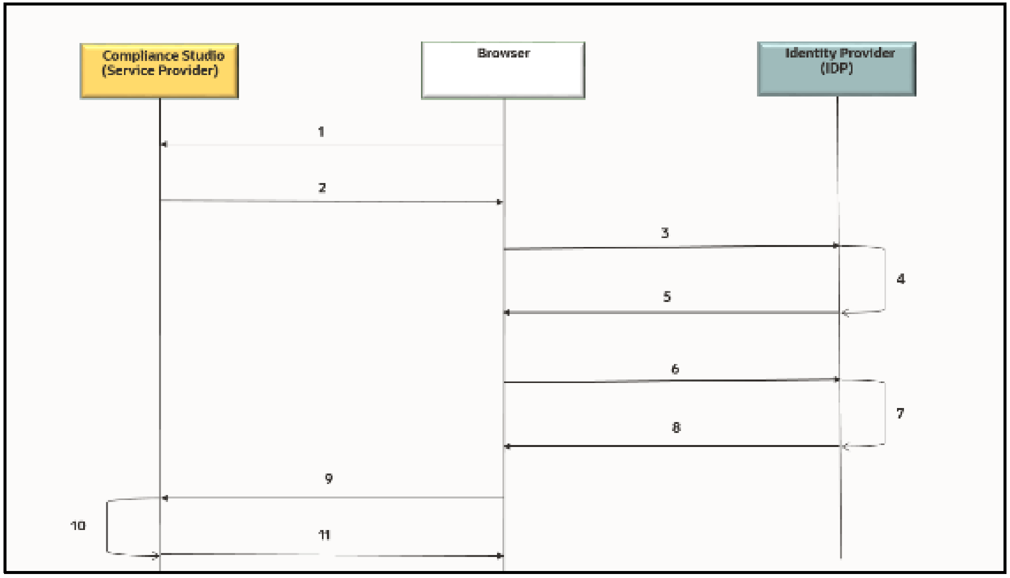
Topics:

- SSO/SAML
- OFSAAI

2.7.1 SSO/SAML

Single Sign-On (SSO)/Security Assertion Markup Language (SAML) is a type of authentication supporting the OFS Compliance Studio. It is an open standard for exchanging authentication and authorization between the user and the Compliance Studio application, such as logins, authentication state, identifiers, and other relevant attributes.

Figure 3: SAML Authentication Process



The entities are as follows:

- End-User
- OFS Compliance Studio application
- SAML

The SAML authentication process is as follows:

1. A user sends a request to access the OFS Compliance Studio application.
2. The application redirects the request to IDP for authentication with SAML request:
3. The application sends the request to IDP for the SSO login page.
4. IDP validates the SAML request for the login page.
5. IDP sends the response to the user with the SSO login page.
6. The user enters the credentials on the SSO login page.
7. IDP validates the credentials and generates the SAML response.
8. IDP sends the SAML response is as follows:
    - For valid credentials, it sends the response to the application for validating the SAML response.
    - For invalid credentials, it displays an authentication error.
9. It posts SAML response to Assertion Consumer URL for valid credentials.
10. The application verifies the user signature in the SAML response.
11. The application displays the OFS Compliance Studio home page to the user.

2.7.2 OFSAAI

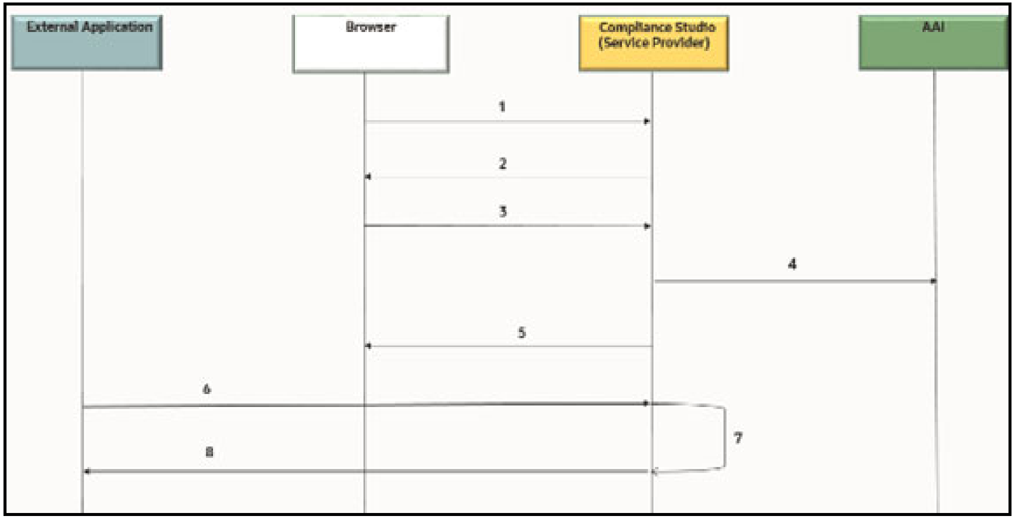
Oracle Financial Services Analytical Applications Infrastructure (OFSAAI) authenticates users using any web browser with a username/password to login into the application. It is also possible to restrict access to content and services based on user attributes or, conversely, make them accessible internationally.

You can authenticate the OFS Compliance Studio with the following:

- Existing OFSAAI
- Install OFSAAI and authenticate

OFSAAI is available with a pre-installed BD Pack or ECM Pack.

Figure 4: OFSAAI Authentication Process



The entities are as follows:

- End-User
- OFS Compliance Studio application
- AAI
- External Application

The AAI authentication process is as follows:

1. A user sends a request to access the OFS Compliance Studio application.
2. The application displays the OFS Compliance Studio application login page:
3. The user enters the credentials on the login page.
4. The application sends the request to AAI for validation.
5. AAI validates the credentials:
    a. For valid credentials, it displays the OFS Compliance Studio home page to the user.
    b. For invalid credentials, it displays an authentication error.
6. The External Application sends the request with Bearer/Basic token to access the application through REST API.
7. The application validates the Authorization Header using Pre-Filters.
8. The application sends the response to the External Application.

    REST API: Representational State Transfer (REST) is a software architectural style that defines a set of constraints to create Web services. Web services that conform to the REST architectural style, called RESTful Web services, provide interoperability between computer systems on the internet.

2.8 Use Cases

This topic provides different use cases.

Topics:

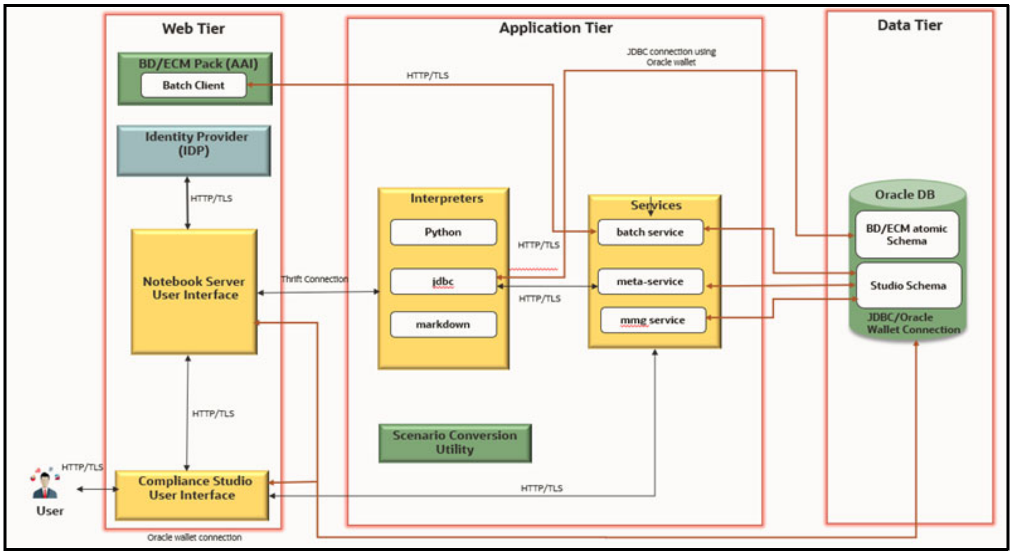
- Scenario Authoring
- Machine Learning for AML
- Entity Resolution
- Investigation Hub

2.8.1 Scenario Authoring

OFS Compliance Studio supports Polyglot Scenario Authoring to author new scenarios in various languages like SQL, Scala, Python, and R.

It is used with Oracle's Behavior Detection or other FCC product. There are pre-built integrations for Scenario Authoring and creating events, posting them into our Enterprise Case Management system, and further creating cases for investigation. However, Compliance Studio can be used with any financial crime platform for Scenario Authoring.

Figure 5: Scenario Authoring



The following components are involved in this use case:

- OFS Compliance Front End Service
- OFS Compliance Back End Service
- IDP
- ECM/BD Pack
- Oracle DB
- Scenario Conversion Utility

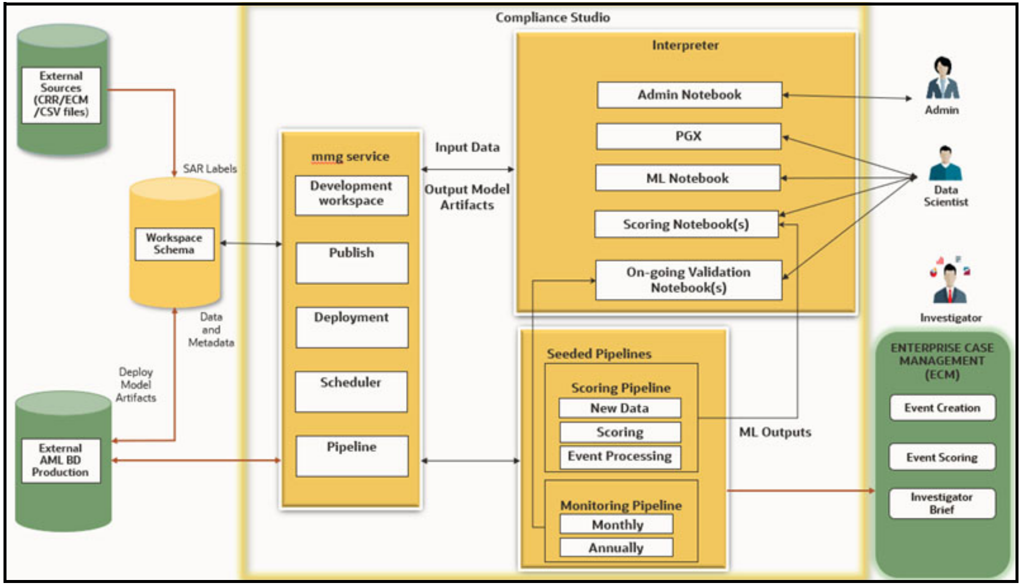
For more information on each component, see the Component Details section.

2.8.2 Machine Learning for AML

OFS Compliance Studio supports Machine Learning for AML (ML4AML). It is a foundation with building blocks for the Machine Learning (ML) lifecycle, tailored for the AML domain. It uses the familiar notebook environment to train, test, and validate ML models rapidly.

It has a predefined dataset with more than 300 attributes ready for variable analysis. You can execute models with multiple techniques and compare the results side-by-side.

Figure 6: ML4AML



The following components are involved in this use case:

- OFS Compliance Front End Service
- OFS Compliance Back End Service
- Database -External sources (ECM/CRR CSV file)/AML BD production
- ECM

For more information on each component, see the Component Details section.

2.8.2.1 Customer Risk Scoring

OFS Compliance Studio supports building the customer risk scoring models from available customer and KYC attributes and behavioral attributes. It uses a set of out-of-the-box behavioral and non-behavioral attributes and Time Series transformations to accelerate feature engineering. You can incorporate AML-related behavioral attributes directly into the model to better assess AML risk.

2.8.2.2 AML Event Scoring

OFS Compliance Studio supports creating an event scoring model that can determine the risk associated with an event. The risk score can be utilized to prioritize events for review or be used as input for case correlation. Leverage alert highlights that are made available OOB besides custom-designed features.

2.8.2.3 Detection Models

OFS Compliance Studio supports building a supervised machine-learning model at an account or customer level to detect the behavior of interest.

2.8.2.4 Customer Segmentation

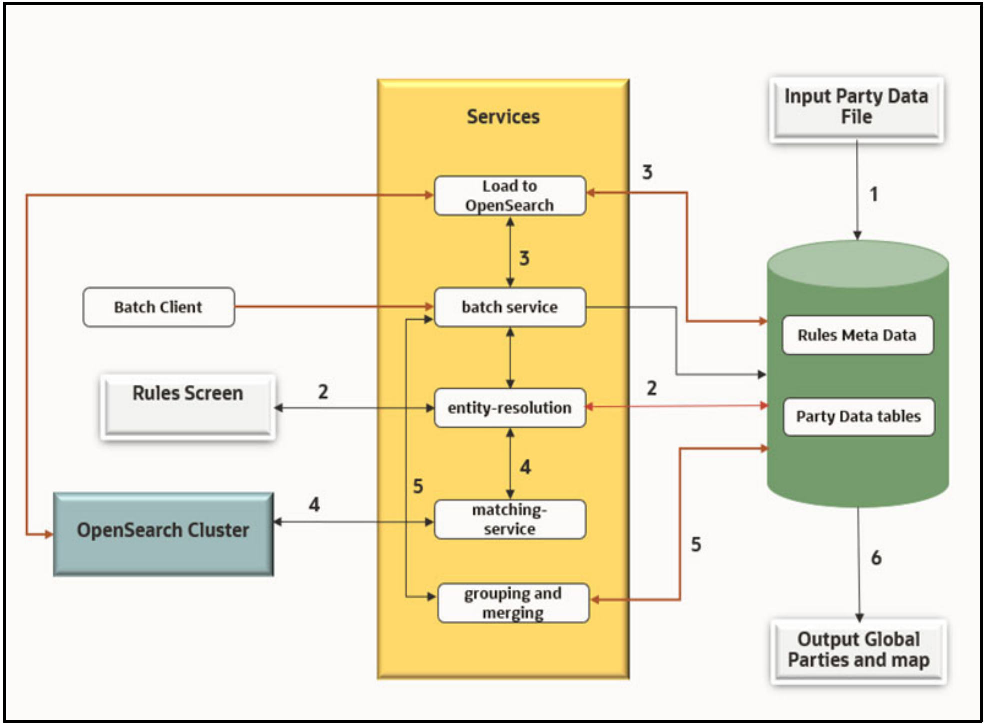
OFS Compliance Studio supports building a top-down customer segmentation framework using the institution's understanding of its risk profile, product portfolio, and customer base. Use unsupervised methods to create bottom-up segments under these higher-level segments. Use anomaly detection techniques to determine if a customer's behavior is inconsistent with that of its segment.

2.8.3 Entity Resolution

OFS Compliance Studio supports Entity Resolution. It allows firms to break through barriers in their data by gaining single views of their customers and their external entities and have the choice of monitoring them both under one consolidated Global Party.

Entity Resolution leverages ideas and concepts from entity resolution, machine learning, and graph analytics to resolve parties across vast datasets where customers, to avoid detection, may misidentify parties due to segmented business processes or malicious attempts. The new features allow firms to have rich visualization around complex networks and truly gain an entity view across varied datasets. This new clear customer view also can be weaponized within AML detection systems by using this resolved data to drive down false positives and ensure entities are being monitored holistically.

Figure 7: Entity Resolution



The following are reference points for Figure 7 :

1. Load input data
2. Input rules
3. Create and load Index
4. Match and generate similarities
5. Group and merge based on similarities
6. Persist Global parties in the file system

The following components are involved in this use case:

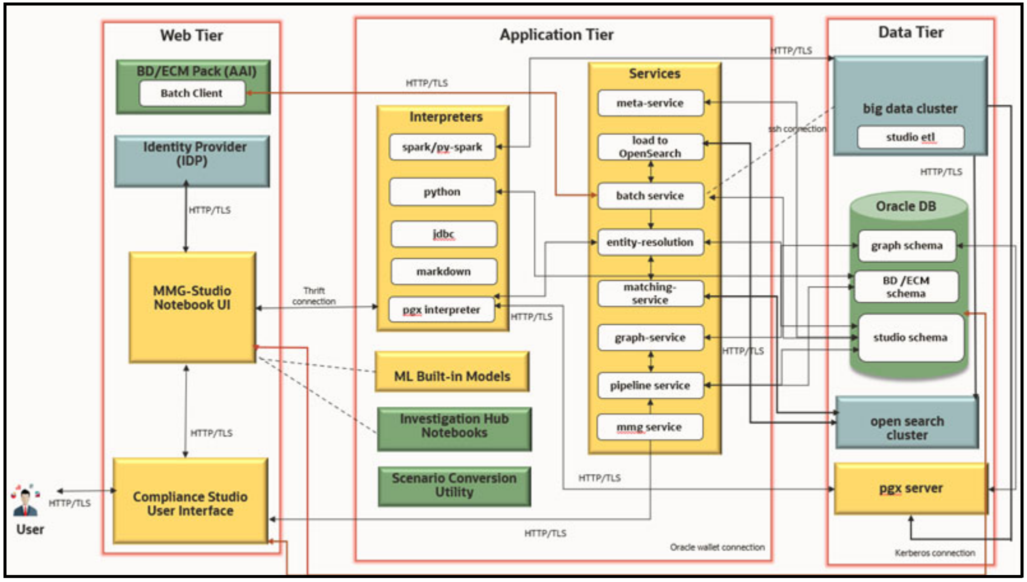
- OFS Compliance Back End Service
- ECM/BD Pack
- Oracle DB
- OpenSearch Cluster

For more information on each component, see the Component Details section.

2.8.4 Investigation Hub

OFS Investigation Hub is an application built on Compliance Studio, allowing investigators to view the case and adhoc information within the FCGM rapidly. The in-built scoring, matching, and correlation engines create meaningful investigation units, and pre-configured red flags and risk factors target investigative efforts effectively. The FCGM on which it is built accelerates investigations by bringing relevant information sources together, preventing the need for the manual collation of information from disparate sources for adhoc investigations. OFS IH automatically generates case narratives and insights, highlights risk factors and red flags meaningful to the investigation, and recommends actions based on graph scoring algorithms.

Figure 8: Investigation Hub



The following components are involved in this use case:

- OFS Compliance Front End Service
- OFS Compliance Back End Service
- IDP
- ECM/BD Pack
- Oracle DB
- OpenSearch
- PGX
- BIG Data Cluster

- Investigation Hub
- Scenario Conversion Utility

For more information on each component, see the Component Details section.

3 High Availability Configuration

The High Availability (HA) architecture is one of the key requirements for any Enterprise Deployment. It refers to the ability of users to access a system without loss of service.

HA preparation is an integral part of contingency planning. This document serves as a reference document for the preparation of specific High Availability (HA) architecture. It explains how a standard Compliance Studio deployment should be architected to protect it from unplanned downtime and minimize planned downtime.

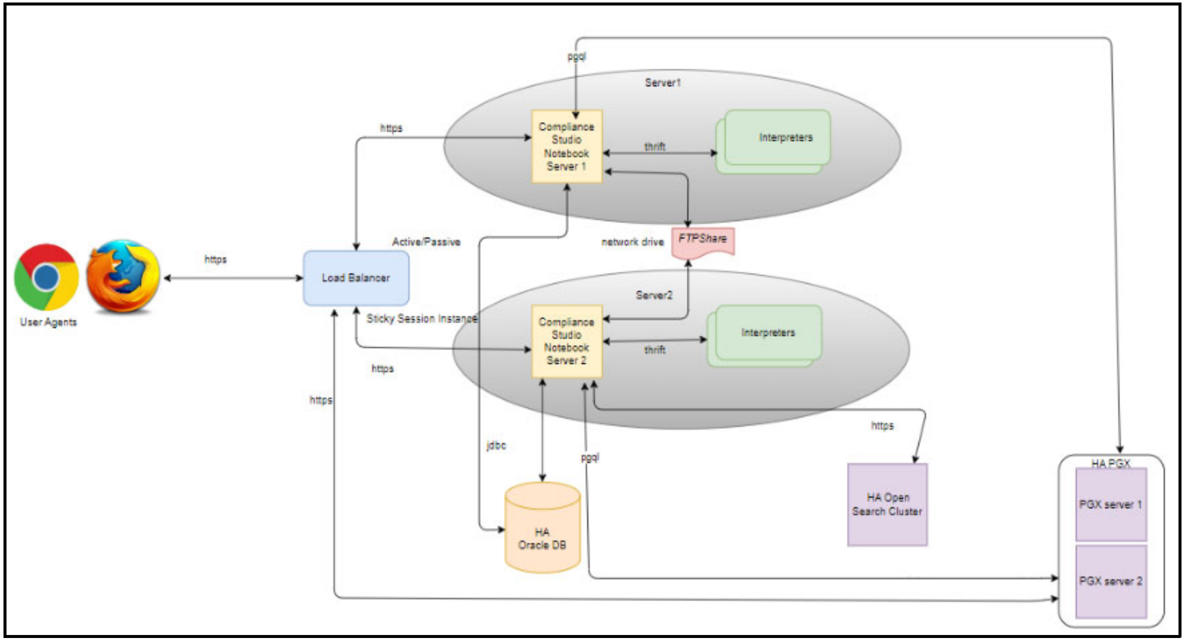
3.1 High Availability Architecture

The high availability of Compliance Studio is currently supported in an **ACTIVE-PASSIVE** mode where only one of the nodes is up at any given time. The HA configuration will need to hit the primary node at all times. When the primary node fails, it requires the secondary node to be fired up manually (the additional manual steps are required, see the section). The manual steps are minimal, and the node can be made active with minimum application downtime.

> **NOTE** The Architecture was validated with HAProxy but other Load Balancers could also be used if needed assuming the same capabilities.

The following illustration shows the HA architecture of the Compliance Studio for active-passive mode.

Figure 9: High Availability Architecture



3.2 Assumptions

- HAProxy is installed in the load balancer server.
- The load balancer, Primary and Secondary Compliance Studio instances should be on the different servers.
- The same schema (Studio, ER/FSDF, Graph, and Atomic) should be used between the Primary and Secondary Compliance Studio instances.

- The same OpenSearch cluster is shared between the Primary and Secondary servers. If different OS clusters are used, then data/indices between these servers should be in sync.
- The certificates (`studio_server.p12` and `studio_server.jks`) generated include the IP addresses/DNS of the load balancer, Primary and Secondary Compliance Studio instances.

3.3 Install Compliance Studio on the Primary Server

To install Compliance Studio on the Primary server, follow these steps:

1. Download the Compliance Studio installer and its associated patches from the My Oracle Support (MOS).

> **NOTE** If the load balancer is enabled for PGX, then the PGX load balancer URL should be provided in the **PGX_SERVER_URL** parameter of the `config.sh` file which is available in the `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/bin` directory.
>
> For example, `PGX_SERVER_URL=http://<PGX_LB_HOSTNAME>:<PGX_LB_PORT>`
>
> This configuration has to be done before installing Compliance Studio on the Primary server and this is required only for the Graph pipeline/PGX server.

2. Install Compliance Studio on the Primary server. For more information on how to install, see the OFS Compliance Studio Installation Guide.

   After successful installation, ensure that all the services are started and the logs are clean.

3. Open the `compliance-studio.sh` file in the following path and update the parameters as mentioned in the Table 6.

Table 6: Parameter of compliance-studio.sh file

File Path	Parameter
`<COMPLIANCE_STUDIO_INSTALLATION_PATH>/bin`	`PGX_INTERPRETER_OPTS="$PGX_INTERPRETER_OPTS -DAPP_BASE_NAME='pgx-interpreter' -Dgraph-service.url=https://<COMPLIANCE_STUDIO_SERVER_IP_ADDRESS>:7059/graph-service`

4. Open the `application.yml` file in the following path and update the parameters as mentioned in the Table 7.

Table 7: Parameter of application.yml file

File Path	Parameter
`<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/mmg-home/mmg-studio/conf`	`mmgserviceUrl=https://<LOAD_BALANCER_HOSTNAME>:7002/cs`

5. Open the `application.properties` file in the following path and update the parameters as mentioned in the Table 8.

Table 8: Parameter of application.properties file

File Path	Parameter
`<COMPLIANCE_STUDIO_INSTALLATION_PATH>/mmg-home/mmg-ui/conf`	`BASE_URL=https://<LOAD_BALANCER_HOSTNAME>:7002/cs`
	`DP_UI_URL=https://<LOAD_BALANCER_HOSTNAME>:7063/pipelineserviceui/pmf/dp/index.jsp`
	`MATCHSRVC_UI_URL=https://<LOAD_BALANCER_HOSTNAME>/fcc/graphmatchruleset.jsp`
	`LOADINDEX_UI_URL=https://<LOAD_BALANCER_HOSTNAME>/fcc/createindex.jsp`
`<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/mmg-home/mmg-pipeline/pipeline/pipelinegateway/conf`	`pipelineservice.uri=https://<LOAD_BALANCER_HOSTNAME>:18005/pipelineservice/`
	`datapipelineservice.uri=https://<LOAD_BALANCER_HOSTNAME>:18006/datapipelineservice`
`<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/mmg-home/mmg-pipeline/pipeline/data-metadata-job-<version>/conf`	`pipeline.url=https://<LOAD_BALANCER_HOSTNAME>:18005/pipelineservice`
`<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/mmg-home/mmg-pipeline/pipeline/data-pipeline-service-<version>/conf`	`mmg.url=https://<LOAD_BALANCER_HOSTNAME>:7002/cs`
	`pipeline.url=https://<LOAD_BALANCER_HOSTNAME>:18005/pipelineservice`
	`datapipeline.url=https://<LOAD_BALANCER_HOSTNAME>:18006/datapipelineservice`
`<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/mmg-home/mmg-pipeline/pipeline/pipeline-service-<version>/conf`	`pipeline.url=https://<LOAD_BALANCER_HOSTNAME>:18005/pipelineservice`
	`datapipeline.url=https://<LOAD_BALANCER_HOSTNAME>:18006/datapipelineservice`
	`gatewayUrl=https://<LOAD_BALANCER_HOSTNAME>:18006`

6. Restart Compliance Studio to reflect the updated configuration. To restart the Compliance Studio on the Primary server, execute the following command.

   ```
   ./compliance-studio.sh --restart
   ```

3.4 Install Compliance Studio on the Secondary Server

To install Compliance Studio on the Secondary server, follow these steps:

> **NOTE** If the load balancer is enabled for PGX, then the PGX load balancer URL should be provided in the **PGX_SERVER_URL** parameter of the `config.sh` file which is available in the `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/bin` directory.
>
> For example, `PGX_SERVER_URL=http://<PGX_LB_HOSTNAME>:<PGX_LB_PORT>`
>
> This configuration has to be done before installing Compliance Studio on the Secondary server and this is required only for the Graph pipeline/PGX server.

1. Compliance Studio can be installed on the Secondary server either as a fresh installation or cloned from the Primary server.
   - Install Compliance Studio (fresh installation) on the Secondary server. For more information on how to install, see the OFS Compliance Studio Installation Guide.
   - For cloning files from the Primary server, see the **Setup Compliance Studio Instance for Cloning the Filesystem** section in the OFS Compliance Studio Installation Guide.

   > **NOTE** Ensure that the following resources are shared/synced between the Primary and Secondary servers for an HA configuration:
   >
   > - Database (Studio, ER/FSDF, Graph, and Atomic schemas)
   > - OpenSearch cluster

   After successful installation/cloning, ensure that all the services are started and the logs are clean.

2. Open the `compliance-studio.sh` file in the following path and update the parameters as mentioned in the Table 9.

Table 9: Parameter of compliance-studio.sh file

File Path	Parameter
`<COMPLIANCE_STUDIO_INSTALLATION_PATH>/bin`	`PGX_INTERPRETER_OPTS="$PGX_INTERPRETER_OPTS -DAPP_BASE_NAME='pgx-interpreter' -Dgraph-service.url=https://<COMPLIANCE_STUDIO_SERVER_IP_ADDRESS>:7059/graph-service`

3. Open the `application.yml` file in the following path and update the parameters as mentioned in the Table 10.

Table 10: Parameter of application.yml file

File Path	Parameter
`<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/mmg-home/mmg-studio/conf`	`mmgserviceUrl=https://<LOAD_BALANCER_HOSTNAME>:7002/cs`

4. Open the `application.properties` file in the following path and update the parameters as mentioned in the Table 11.

Table 11: Parameter of application.properties file

File Path	Parameter
<COMPLIANCE_STUDIO_INSTALLATION_PATH>/mmg-home/mmg-ui/conf	BASE_URL=https://<LOAD_BALANCER_HOSTNAME>:7002/cs
	DP_UI_URL=https://<LOAD_BALANCER_HOSTNAME>:7063/pipelineserviceui/pmf/dp/index.jsp
	MATCHSRVC_UI_URL=https://<LOAD_BALANCER_HOSTNAME>/fcc/graphmatchruleset.jsp
	LOADINDEX_UI_URL=https://<LOAD_BALANCER_HOSTNAME>/fcc/createindex.jsp
<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/mmg-home/mmg-pipeline/pipeline/pipelinegateway/conf	pipelineservice.uri=https://<LOAD_BALANCER_HOSTNAME>:18005/pipelineservice/
	datapipelineservice.uri=https://<LOAD_BALANCER_HOSTNAME>:18006/datapipelineservice
<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/mmg-home/mmg-pipeline/pipeline/data-metadata-job-<version>/conf	pipeline.url=https://<LOAD_BALANCER_HOSTNAME>:18005/pipelineservice
<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/mmg-home/mmg-pipeline/pipeline/data-pipeline-service-<version>/conf	mmg.url=https://<LOAD_BALANCER_HOSTNAME>:7002/cs
	pipeline.url=https://<LOAD_BALANCER_HOSTNAME>:18005/pipelineservice
	datapipeline.url=https://<LOAD_BALANCER_HOSTNAME>:18006/datapipelineservice

Table 11: Parameter of application.properties file

File Path	Parameter
<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/mmg-home/mmg-pipeline/pipeline/pipeline-service-<version>/conf	pipeline.url=https://<LOAD_BALANCER_HOSTNAME>:18005/pipelineservice
	datapipeline.url=https://<LOAD_BALANCER_HOSTNAME>:18006/datapipelineservice
	gatewayUrl=https://<LOAD_BALANCER_HOSTNAME>:18006

5. Restart Compliance Studio to reflect the updated configuration. To restart the Compliance Studio on the Secondary server, execute the following command.

```
./compliance-studio.sh --restart
```

3.5 Studio Schema Configuration

To configure the Studio schema, follow these steps:

1. Log in to Studio schema.
2. Open the **NEXTGENEMF_CONFIG** table and update value in the **V_VALUE** column as described in the Table 12.

Table 12: Parameter for NEXTGENEMF_CONFIG

V_NAME	V_VALUE
EMFSTUDIO_SERVICE_URL	https://<LOAD_BALANCER_HOSTNAME>:7002/cs
BASE_URL	https://<LOAD_BALANCER_HOSTNAME>:7002/cs
DATASTUDIO_URL	https://<LOAD_BALANCER_HOSTNAME>:7008/cs

3. Open the **MMG_MENU** table and update value in the **V_MENU_URL** column as described in the Table 13.

Table 13: Parameter for MMG_MENU Table

V_MENU_NAME	V_MENU_URL
Match Rules	https://<LOAD_BALANCER_HOSTNAME>/fcc/matchrulesetsummary.jsp
Merge Rules	https://<LOAD_BALANCER_HOSTNAME>/fcc/mergerulesetsummary.jsp
Data Survival	https://<LOAD_BALANCER_HOSTNAME>/fcc/datasurvivalsummary.jsp

Table 13: Parameter for MMG_MENU Table

V_MENU_NAME	V_MENU_URL
Manual Decisioning	https://<LOAD_BALANCER_HOSTNAME>/fcc/manualdecisioning.jsp
Merge and Split Global Entities	https://<LOAD_BALANCER_HOSTNAME>/fcc/mergeandsplit.jsp
Data Pipelines	https://<LOAD_BALANCER_HOSTNAME>:7063/pipelineserviceui/pmf/dp/index.jsp

4. Open the **AAICL_SS_BATCH_URL** table and update value in the **V_URL** column as described in the Table 14.

Table 14: Parameter for MMG_MENU Table

V_URL_NAME	V_URL
MMG_SERVICE_URL	https://<LOAD_BALANCER_HOSTNAME>:7002/cs
WORKSPACE_URL	https:// <LOAD_BALANCER_HOSTNAME>:7002/cs
CS_SERVICE_URL	https:// <LOAD_BALANCER_HOSTNAME>:7002/cs

5. Restart Compliance Studio after making changes to the Studio schema. To restart, execute the following command.

```
./compliance-studio.sh --restart
```

3.6 Switching from One Server to Another Server

Users can switch from Primary server to Secondary server and vice versa at any time.

> **NOTE** Ensure that all the Compliance Studio services from the other server are down.

To switch from one server to another, follow these steps:

1. Delete entry from the Studio schema using the following query.

```
select * from DATABASECHANGELOG where author = 'Compliance Studio
8.1.2.1' and id   =   'FCC_DATASTUDIO_CONFIG_8121';
```

2. Start Compliance Studio on the other server. To start, execute the following command.

```
./compliance-studio.sh --start
```

3.7 Manual Configurations for Each Use Case

Users need to configure manually for the following use cases based on their requirement.

3.7.1 Entity Resolution

To configure the entity resolution, follow these steps:

1. Navigate to `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/ficdb/conf` directory.
2. Open the `resources.xml` file and update the following ER/FSDF schema details.

```
<Resource

id="<ER_SCHEMA_ALIAS>"

name="jdbc/erdataschema"

auth="Container"

type="javax.sql.DataSource"

driverClassName="oracle.jdbc.OracleDriver"

url="jdbc:oracle:thin:@<ER_SCHEMA_ALIAS>"
connectionProperties="oracle.net.wallet_location=<STUDIO_WALLET_LOCAT
ION>;oracle.net.tns_admin=<STUDIO_TNS_ADMIN_PATH>;"

maxTotal="20"

maxIdle="0"

maxWaitMillis="-1">

</Resource>
```

3.7.2 Interpreters

Interpreters must be configured in both servers independently to work when switching from one server to another.

To configure Interpreters, see the **Configure Interpreters** section in the OFS Compliance Studio Administration and Configuration Guide.

3.7.3 PGX Server

Users need to configure both Compliance Studio and PGX server when load balancer is enabled.

3.7.3.1 In Compliance Studio

If the load balancer is enabled for PGX, then the PGX load balancer URL should be provided in the **PGX_SERVER_URL** parameter of the `config.sh` file which is available in the `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/bin` directory.

For example, `PGX_SERVER_URL=http://<PGX_LB_HOSTNAME>:<PGX_LB_PORT>`

This configuration has to be done before installing Compliance Studio on the Primary and Secondary servers.

3.7.3.2 In PGX Server

Prerequisites: Load Balancer setup is required for HAProxy.

For configuring and installing PGX Server, see the Compliance Studio with Multiple PGX Servers (Using Load Balancer) section.

3.7.4 Scenario Conversion Utility

Before running the scenario generation notebook, update the **obj_url** and **url** variables in the **Generate Scenario(s)** paragraph to point to the respective Compliance Studio server that is currently running.

For more information, see the **Using Scenario Conversion Utility** section in the OFS Compliance Studio User Guide.

3.7.5 Data Pipelines

The pem format of the studio_server certificate is referred to in the frontend configuration in the load balancer configuration.

Example:

```
frontend datapipeline_service

      bind *:18006 ssl crt /etc/ssl/certs/haproxy.pem

     mode http

     default_backend datapipeline_service
```

Here, `haproxy.pem` is the pem format of the `studio_server.p12` file that is available in the `/etc/ssl/certs/` path.

3.8 HAProxy Configuration

To configure HAProxy, follow these steps:

1. Navigate to `/etc/haproxy` directory and execute following command in the terminal.

   ```
   cd /etc/haproxy
   ```

2. Open the `haproxy.cfg` file and execute following command in the terminal.

   ```
   vi haproxy.cfg
   ```

3. Add the following frontend and backend blocks in the `haproxy.cfg` file.

# HAPROXY CONFIGURATION STARTS

#---------------------------Frontend configurations-------------------------------------

```
frontend compliance_studio

   bind *:7001

   mode http

   bind *:443 ssl crt /etc/ssl/certs/haproxy.pem

   use_backend fcc_ui if { path /fcc } || { path_beg /fcc/ }

   http-request set-header X-Forwarded-For %[src]

   reqadd X-Forwarded-Proto:\ https

   option http-server-close

   default_backend compliance_studio
```

```
frontend mmg
   bind *:7002 ssl crt /etc/ssl/certs/haproxy.pem
   mode http
   default_backend mmg

frontend data_studio
   bind *:7008 ssl crt /etc/ssl/certs/haproxy.pem
   mode http
   default_backend data_studio

frontend dp_ui
   bind *:7063 ssl crt /etc/ssl/certs/haproxy.pem
   mode http
   default_backend dp_ui

frontend graph_service
   bind *:7059 ssl crt /etc/ssl/certs/haproxy.pem
   mode http
   default_backend graph_service

frontend pipeline_service
   bind *:18005 ssl crt /etc/ssl/certs/haproxy.pem
   mode http
   default_backend pipeline_service

frontend datapipeline_service
   bind *:18006 ssl crt /etc/ssl/certs/haproxy.pem
   mode http
   default_backend datapipeline_service

frontend pgx_server
   bind *:7007
   mode http
   default_backend pgx_server
```

```
#----------------------------Backend  configurations------------------------------------------

    backend compliance_studio
       mode http
       balance roundrobin
       cookie JSESSIONID prefix nocache
       server cs_server_1 <server_1_hostname>:7001/cs/home check ssl verify
       none
       server cs_server_2 <server_2_hostname>:7001/cs/home check ssl verify
       none

    backend mmg
       mode http
       balance roundrobin
       cookie JSESSIONID prefix nocache
       server mmg_1 <server_1_hostname>:7002 check ssl verify none
       server mmg_2 <server_2_hostname>:7002 check ssl verify none

    backend data_studio
       mode http
       balance roundrobin
       cookie JSESSIONID prefix nocache
       server data_studio_1 <server_1_hostname>:7008/cs check ssl verify
       none
       server data_studio_2 <server_2_hostname>:7008/cs check ssl verify
       none

    backend dp_ui
       mode http
       balance roundrobin
       cookie JSESSIONID prefix nocache
       server dp_ui_1 <server_1_hostname>:7063/pipelineserviceui check ssl
       verify none
       server dp_ui_2 <server_2_hostname>:7063/pipelineserviceui check ssl
       verify none
```

```
backend fcc_ui
   mode http
   balance roundrobin
   cookie JSESSIONID prefix nocache
   server fcc_ui_1 <server_1_hostname>:7061/fcc check ssl verify none
   server fcc_ui_2 <server_2_hostname>: 7061/fcc check ssl verify none

backend graph_service
   mode http
   balance roundrobin
   cookie JSESSIONID prefix nocache
   server graph_service_1 <server_1_hostname>:7059 check ssl verify none
   server graph_service_2 <server_2_hostname>:7059 check ssl verify none

backend pipeline_service
   mode http
   balance roundrobin
   cookie JSESSIONID prefix nocache
   server pipeline_service_1 <server_1_hostname>:18005 check ssl verify
   none
   server pipeline_service_2 <server_2_hostname>:18005 check ssl verify
   none

backend datapipeline_service
   mode http
   balance roundrobin
   cookie JSESSIONID prefix nocache
   server datapipeline_service_1 <server_1_hostname>:18006 check ssl
   verify none
   server datapipeline_service_2 <server_2_hostname>:18006 check ssl
   verify none

backend pgx_server
   mode http
   cookie PGX_INSTANCE_STICKY_COOKIE insert indirect nocache
   server pgx_server_1 <server_1_hostname>:7007 check
   server pgx_server_2 <server_1_hostname>:7007 check cookie
   pgx_server_2
```

```
    option httpchk GET /isReady
    http-check expect string true
```

# HAPROXY CONFIGURATION ENDS

4 Compliance Studio with Multiple PGX Servers (Using Load Balancer)

> **NOTE** This section describes HAProxy but other load balancers can be used.

Using a load balancer, users can configure multiple PGX servers in the Compliance Studio.

HAProxy is a high-performance TCP/HTTP load balancer and proxy server that allows multiplexing incoming requests across multiple web servers. You can use HAProxy with multiple instances of the graph server (PGX) for high availability.

Prerequisites

- Load Balancer setup is required for HAProxy.
- Two or more servers should be available for the PGX server.

To configure multiple PGX servers, follow these steps:

1. Configure and Install PGX on both servers. To Configure and Install PGX, see the **Configure the PGX Service** section in the OFS Compliance Studio Installation Guide.
2. Navigate to `<PGX_SERVER_HOME>/pgx-server/conf` directory and place the following files:
   - `studio_server.p12`
   - `public.key` and `private.key`
   - `graph-keystore.p12`

> **NOTE**
> - For more information on where to obtain `studio_server.p12` file, see the section in the OFS Compliance Studio Installation Guide.
> - For more information on where to obtain `public.key` and `private.key` files, see the section in the OFS Compliance Studio Installation Guide.
> - For more information on where to obtain `graph-keystore.p12` file, see the section in the OFS Compliance Studio Installation Guide.

3. Start the PGX server.
4. Install the Load balancer HAProxy in one of the PGX servers as mentioned in the Using HAProxy for PGX Load Balancing and High Availability section.
5. Navigate to `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/bin` directory.
6. Open the `config.sh` file and provide a Load balancer URL in the **PGX_SERVER_URL** parameter.

```
PGX_SERVER_URL=http://<PGX_LB_Hostname>:<PGX_LB_PORT>
```

For example, `PGX_SERVER_URL=http://testserver.oracle.com/:1234`

NOTE	- The PGX load balancer can be configured in the same server that is used for load balancing the Compliance Studio as well. - Only Subgraph loading is supported and In-memory graph loading is not supported if HA is configured for the PGX server.

7. Start Compliance Studio. To start, execute the following command.

```
./compliance-studio.sh --start
```

5 Setup Disaster Recovery (DR) in Compliance Studio

This section describes additional configuration required to setup Disaster Recovery (DR) in the Compliance Studio.

Prerequisites:

- Create a wallet. To create a wallet, see the **Setup the Password Stores for Database User Accounts** section in the OFS Compliance Studio Installation Guide.
- Add Secondary Database credentials in the wallet.
- ER/FSDF Schema, Atomic Schema, Studio Schema, and Graph Schema names should be the same as the Primary Database.

To setup DR in the Compliance Studio, follow these steps:

1. Navigate to the `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/bin` directory.
2. Open `config.sh` file and update the parameters as mentioned in Table 15 .

Table 15: Parameter Values in Config.sh File

Parameter	Placeholder Value
DB Details of Studio Schema	
STUDIO_DB_HOSTNAME	## SECONDARY_STUDIO_DB_HOSTNAME ##
STUDIO_DB_PORT	## SECONDARY _STUDIO_DB_PORT##
STUDIO_DB_SERVICE_NAME	## SECONDARY _STUDIO_DB_SERVICE_NAME ##
STUDIO_DB_SID	## SECONDARY _STUDIO_DB_SERVICE_NAME ##
STUDIO_DB_USERNAME	For example, CS81250_DR_2477
DB Details of Atomic Schema	
ATOMIC_DB_HOSTNAME	## SECONDARY _ATOMIC_DB_HOSTNAME ##
ATOMIC_DB_PORT	## SECONDARY _STUDIO_DB_PORT##
ATOMIC_DB_SERVICE_NAME	## SECONDARY _ATOMIC_DB_SERVICE_NAME ##
ATOMIC_DB_SID	## SECONDARY _ATOMIC_DB_SERVICE_NAME ##
ATOMIC_DB_USERNAME	For example, STD_ATOM8125
Graph Schema Configuration	
GRAPH_DB_SERVER_NAME	## SECONDARY _GRAPH_DB_SERVER_NAME ##
GRAPH_DB_PORT	## SECONDARY_GRAPH_DB_PORT ##
GRAPH_DB_SERVICE_NAME	## SECONDARY _GRAPH_DB_SERVICE_NAME ##

Table 15: Parameter Values in Config.sh File

Parameter	Placeholder Value
GRAPH_KEYSTORE_PASSWORD	For example, password123
GRAPH_SCHEMA_DB_SCHEMA_NAME	For example, GS81250_DR_2477
GRAPH_SCHEMA_WALLET_ALIAS	For example, GS81250_DR_2477_alias
GRAPH_SCHEMA_WALLET_LOCATION	##POINTING_TO_DR_DB##
GRAPH_SCHEMA_TNS_ADMIN_PATH	##POINTING_TO_DR_DB##
Wallet Details	
WALLET_LOCATION	##POINTING_TO_DR_DB##
TNS_ADMIN_PATH	##POINTING_TO_DR_DB##

> **NOTE** For the parameter description, see the **Configure the config.sh File** section in the OFS Compliance Studio Installation Guide.

3. Stop the Compliance Studio by executing the following command:

```
./compliance-studio.sh --stop
```

4. Reinstall the Compliance Studio by executing the following command:

```
./compliance-studio.sh --reinstall
```

5. Start the Compliance Studio by executing the following command:

```
./compliance-studio.sh --start
```

6. Navigate to the `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/ficdb/conf` directory.

7. Open the `resources.xml` file and update the following details for ER/FSDF schema.

Example:

```
<Resource
   id="###ER_SCHEMA#"
   name="jdbc/erdataschema"
   auth="Container"
   type="javax.sql.DataSource"
   driverClassName="oracle.jdbc.OracleDriver"
   url="jdbc:oracle:thin:@ "###ER_SCHEMA#""
   connectionProperties= "oracle.net.wallet_location
   =<WALLET_PATH/ABCD>;
   oracle.net.tns_admin=<WALLET_PATH/ABCD>;”
   maxTotal="5"
   maxIdle="0"
```

```
maxWaitMillis="-1" >
</Resource>
```

6 Frequently Asked Questions (FAQs)

This section consists of resolutions to the frequently asked questions noticed during the Compliance Studio installation.

1. What should I do if the data pipelines are failing with the following error?

   ```
   09/Jun/2022 10:48:32,978- ImportDAOImpl: Exception in import processor

   org.springframework.web.client.ResourceAccessException: I/O error on
   POST request for "https://<LOAD_BALANCER_HOSTNAME>:18004/
   datapipelineservice/MAP/IMPORT": PKIX path building failed:
   sun.security.provider.certpath.SunCertPathBuilderException: unable to
   find valid certification path to requested target; nested exception is
   javax.net.ssl.SSLHandshakeException: PKIX path building failed:
   sun.security.provider.certpath.SunCertPathBuilderException: unable to
   find valid certification path to requested target

   at
   org.springframework.web.client.RestTemplate.doExecute(RestTemplate.java:
   748) ~[spring-web-5.2.5.RELEASE.jar!/:5.2.5.RELEASE]

   at
   org.springframework.web.client.RestTemplate.execute(RestTemplate.java:67
   4) ~[spring-web-5.2.5.RELEASE.jar!/:5.2.5.RELEASE]

   at
   org.springframework.web.client.RestTemplate.postForObject(RestTemplate.j
   ava:418) ~[spring-web-5.2.5.RELEASE.jar!/:5.2.5.RELEASE]

   at
   com.oracle.fccm.amlxe.pipelineService.client.RESTClient.callRESTService(
   RESTClient.java:107) ~[clas
   ```

   To resolve this issue, Ensure that the pem format of the studio_server certificate is referred to in the frontend configuration in the load balancer configuration.

   Example:

   ```
   frontend datapipeline_service

      bind *:18006 ssl crt /etc/ssl/certs/haproxy.pem

      mode http

      default_backend datapipeline_service
   ```

   Here, `haproxy.pem` is the pem format of the `studio_server.p12` file that is available in the `/etc/ssl/certs/` path.

2. What should I do if Compliance Studio server fails with the below error in `metaservice.log` when switching from one server to another?

   ```
   26-03-2024 08:35:26.942 [ main] ERROR
   ofss.fccm.applicationserver.server.GrizzlyServer - Server failed to
   start liquibase.exception.ValidationFailedException: Validation Failed:
   1 change sets check sum model/datamodel/
   fccstudio_Atomic_Schema_8121.xml::FCC_DATASTUDIO_CONFIG_8121::Compliance
   Studio 8.1.2.1 was: 8:34bcb65a5125e53bd31acbc46a504d04 but is now:
   8:c52d8dba1a04bcd378a9744484ab01be at
   liquibase.changelog.DatabaseChangeLog.validate(DatabaseChangeLog.java:29
   6) ~[liquibase-core-4.8.0.jar:?]
   ```

To resolve this issue, before switching the Compliance Studio instance from one server to another delete the record returned by the following query.

```
select * from DATABASECHANGELOG where author = 'Compliance Studio
8.1.2.1' and id = 'FCC_DATASTUDIO_CONFIG_8121';
```

3. What should I do if PGX fails to start/restart and displays the below error in the `pgx-server.log`?

```
22-05-2024 08:48:33 ERROR o.p.r.PgxContextListener - Exception while
initializing PGX webapp

java.util.concurrent.ExecutionException:
java.lang.IllegalArgumentException: javax.net.ssl.SSLHandshakeException:
No name matching <PGX_SERVER_HOSTNAME> found

at java.base/
java.util.concurrent.CompletableFuture.reportGet(CompletableFuture.java:
395)

    at java.base/
java.util.concurrent.CompletableFuture.get(CompletableFuture.java:1999)

    at oracle.pgx.api.PgxFuture.get(PgxFuture.java:113)

    at
oracle.pgx.rest.PgxContextListener.contextInitialized(PgxContextListener
.java:64)

    at
org.apache.catalina.core.StandardContext.listenerStart(StandardContext.j
ava:4462)

    at
org.apache.catalina.core.StandardContext.startInternal(StandardContext.j
ava:4914)

    at
org.apache.catalina.util.LifecycleBase.start(LifecycleBase.java:171)

    at
org.apache.catalina.core.ContainerBase$StartChild.call(ContainerBase.jav
a:1332)

    at
org.apache.catalina.core.ContainerBase$StartChild.call(ContainerBase.jav
a:1322)

    at java.base/
java.util.concurrent.FutureTask.run(FutureTask.java:264)


.. .. ..


Caused by: java.lang.IllegalArgumentException:
javax.net.ssl.SSLHandshakeException: No name matching
<PGX_SERVER_HOSTNAME> found

    at oracle.pgx.engine.admin.Ctrl.preloadGraphs(Ctrl.java:355)
    at oracle.pgx.engine.admin.Ctrl.access$1600(Ctrl.java:97)
    at oracle.pgx.engine.admin.Ctrl$1.call(Ctrl.java:248)
    at oracle.pgx.engine.admin.Ctrl$1.call(Ctrl.java:186)
    at
oracle.pgx.api.admin.internal.AbstractControl.runOnCallerThread(Abstract
Control.java:45)
    at oracle.pgx.engine.admin.Ctrl.start(Ctrl.java:186)
    at
oracle.pgx.api.admin.internal.AbstractControl.start(AbstractControl.java
:115)
    at
oracle.pgx.api.admin.internal.AbstractControl.lambda$start$5(AbstractCon
trol.java:101)
    at java.base/
java.util.function.Function.lambda$andThen$1(Function.java:88)
    at java.base/
java.util.concurrent.CompletableFuture.uniComposeStage(CompletableFuture
.java:1106)
    at java.base/
java.util.concurrent.CompletableFuture.thenCompose(CompletableFuture.jav
a:2235)
    at oracle.pgx.api.PgxFuture.thenCompose(PgxFuture.java:178)
    at
oracle.pgx.api.admin.internal.AbstractControl.start(AbstractControl.java
:101)
    at
oracle.pgx.api.admin.internal.AbstractControl.lambda$start$1(AbstractCon
trol.java:79)
    at java.base/
java.util.function.Function.lambda$andThen$1(Function.java:88)
    at java.base/
java.util.concurrent.CompletableFuture.uniComposeStage(CompletableFuture
.java:1106)
    at java.base/
java.util.concurrent.CompletableFuture.thenCompose(CompletableFuture.jav
a:2235)
    at oracle.pgx.api.PgxFuture.thenCompose(PgxFuture.java:178)
    at
oracle.pgx.api.admin.internal.AbstractControl.start(AbstractControl.java
:67)
    at
oracle.pgx.rest.PgxContextListener.contextInitialized(PgxContextListener
.java:62)
    ... 26 common frames omitted
```

This happens due to graph was loaded with IN-MEMORY mode which is not supported in a HA PGX configuration.

To clean up the IN-MEMORY graph and reload it in OFFLOADED mode, follow these steps:

a. Truncate the records from **FCC_PGX_M_CONFIG**, **FCC_GRAPH_M_DATA_SOURCES** and **FCC_GRAPH_M_CONFIG_JSON** tables.

b. Start the PGX servers.

c. Run the **Refresh Graph** task with graph type as **OFFLOADED**.

OFSAA Support

Raise a Service Request (SR) in My Oracle Support (MOS) for queries related to the OFSAA applications.

Send Us Your Comments

Oracle welcomes your comments and suggestions on the quality and usefulness of this publication. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most about this manual?

If you find any errors or have any other suggestions for improvement, indicate the title and part number of the documentation along with the chapter/section/page number (if available) and contact the Oracle Support.

Before sending us your comments, you might like to ensure that you have the latest version of the document wherein any of your concerns have already been addressed. You can access My Oracle Support site which has all the revised/recently released documents.